2.0 CONCEPTUAL MODEL SPECIFICATION

This Conceptual Model Specification (CMS) has been compiled from several Post-Development Design Document (PD3) sections and is intended to be a substitute for the Software Design Document (SDD) called for in DOD-STD-2167A. *ESAMS* is a mature model written before these standards were in place and therefore does not have an SDD. The CMS delineates both high-level and detailed design descriptions and requirement specifications for *ESAMS*. Thus, it serves as the basis for model verification. The progression of conceptual descriptions from the very general (operational concept) to the very specific (detailed design) should also facilitate comprehension of the *ESAMS* model.

The CMS is being developed in stages, and is incomplete. The current stage includes operational concept and high-level design information, as well as detailed design for several FEs. Additional FE detailed design sections will be incorporated as changes to this document as the model continues to evolve.

Operational concept, design, and requirements specifications are being determined from several key sources. Top-level mission and system design information was consolidated from all available model developer documentation such as user's, analyst's, and programmer's manuals. The primary sources for detailed design information are the Verification Source Reports (VSRs) produced by the model developer for the SMART Project. The VSRs provide detailed information about code structure as well as sources and design of algorithms. Additional information has been obtained from Computer-Aided Software Engineering (CASE) tool analyses and direct code examination.

ESAMS has been subdivided into functional areas (top-level functions) which have been further subdivided into functional elements (FEs) that essentially represent individual subsystems and/or functions of the physical systems modeled. A Functional Area Template (FAT) with numbered FEs was developed to provide a frame of reference for models having similar functions and is included in Appendix A.

2.0.1 OPERATIONAL CONCEPT

Mission Of Model

ESAMS is a package of digital computer models designed to simulate encounter(s) between a single airborne target and a surface-to-air missile (SAM) system. It provides a one-on-one framework in which to evaluate air vehicle survivability as a function of tactics, vehicle characteristics, and countermeasures. *ESAMS* models primary elements of a SAM engagement, including sensor lock-on and tracking, missile flight dynamics, missile guidance and control, offensive/defensive countermeasures, and endgame (warhead/fuzing). Though the primary model result is probability of target kill, the *ESAMS* user can examine details of other aspects such as the missile flight path, guidance characteristics, and the effect of electronic countermeasures (ECM) and terrain on an engagement.

The primary mission areas of the model are: (1) to provide data to assess the survivability of an airborne target against a surface-to-air missile (SAM) system, and (2) to aid analysts in the study of SAM engagement phenomena. A secondary objective of *ESAMS* is to generate output data for input to higher-order models that predict mission and/or campaign level results.

ESAMS has been used in the following areas of user interest:

- survivability analysis
- tactics development
- countermeasure effectiveness
- attrition analysis
- support for force structure studies
- research and development inputs for new systems

2.0.2 OPERATIONAL CAPABILITIES

ESAMS simulates weapon subsystems and some electronic countermeasures (ECM) for the following SAM systems:

SA-2	SA-8	SA-14	SA-N-6
SA-3	SA-9	SA-15	SA-N-7
SA-4	SA-10	SA-16	SA-N-9
SA-5	SA-11	SA-N-1	CADS-1
SA-6	SA-12	SA-N-3	
SA-7	SA-13	SA-N-4	

ESAMS has the capability to generate a simple flight path internally or accept flight paths created by Blue Max II, an external flight path generator. Flight path data from testing may also be used via this latter option.

2.0.3 IMPLICATIONS FOR MODEL USE

ESAMS has the capability to generate a variety of outputs that can be used in various aspects of operational and effectiveness analysis. Because of the type of phenomena and level of detail simulated, the model is usually installed on powerful computer systems, but execution can still require hours of processing time. Radio frequency (RF) systems are modeled at greater levels of detail than infrared (IR) systems and therefore require more computing time per engagement. Due to the type of systems simulated, a classified processing environment is also required and the user must provide appropriate security clearance information in order to receive a copy of the software.

While models of some systems have been available for years, others continue to be added and improved via development and maintenance efforts of the model manager. In recent years, a concerted effort to improve ECM capabilities has been made, although funding and data necessary for their validation has not been available. This is a common problem associated with models of this type, which are continually enhanced with new features and capabilities before V&V efforts can be applied to them. In addition, production of updated documentation has been sacrificed for software development.

Software design of *ESAMS* components and functions is sophisticated, although quite modular, and many routines are applicable to classes of systems, which facilitates both upgrades and maintenance. Radar function sensitivities can be overshadowed by those having to do with missile flight and end game calculations of kill probabilities, but they are the only ones that have been examined and reported on thus far.

2.0.4 TOP-LEVEL SOFTWARE DESIGN

Major Components

An air defense unit of the type represented by *ESAMS* is designed to protect a defended area from air penetration. The target is defined by its flight path (position, velocity, and attitude as a function of time), its reflectance and its vulnerability. In each ground-to-air interaction, the SAM site attempts to acquire and lock onto the target with its target tracking radar (TTR). The minimum signal required for acquisition, track establishment, and track as measured by the SAM radar is a function of many factors (atmospheric transmittance, multipath and clutter, terrain masking, etc.), and countermeasure interference.

Major functional areas of *ESAMS* are the RF sensor, missile flyout, and end game, but the SMART Project has limited examination to the first two because of their association with susceptibility. The RF sensor includes acquisition and tracking modes of the target tracking radar while missile flyout includes launch/no-launch decision criteria as well as flight dynamics and control. Infrared (IR) sensor capability and end game (i.e., fuzing and warhead) functions have not been addressed under SMART Project efforts.

RF Sensor Acquisition Mode

The *ESAMS* simulation starts with the tracking radar in acquisition mode. The sensor is “perfectly cued” to the target. In this mode the radar antenna is kept pointing directly at the flying target until its signal is detected above the noise and clutter levels. This does not correspond to a normal operational radar system mode. This mode determines the longest range at which the target can be detected.

RF Sensor Track Mode

After successful acquisition of the target penetrator, the sensor then switches to the track mode of the tracking radar. The radar attempts to track the target in azimuth, elevation, and range. The radar is still “staring” at the target penetrator as it was in the acquisition mode. Normally, this mode generates a target lock-on which serves as the transition to the missile flyout portion of the simulation.

Missile Flyout

The flyout of a missile is preceded by acquisition or track establishment phases, or both. Countermeasures may be invoked during these phases. Before commencing its firing sequence, the site must meet specified launch criteria, including target range, velocity, and signal-to-noise ratio (SNR). Meeting these criteria, the site begins its firing sequence and launches a single missile. For a short time the missile boosts, then guidance begins. Ground or onboard commands are generated to guide the missile toward the target.

The missile is then guided to close with the target until fuzing begins. During missile flyout, the model provides for simulation of user-selectable instrumental and environmental noise processes, as well as the effects of various types of countermeasures.

Upon fuzing, the model calculates the probabilities-of-kill from both warhead blast and fragmentation. The probabilities are based on warhead fragment characteristics, including fragment size and velocity, the aircraft vulnerability to the fragments, and the aircraft vulnerability to the warhead blast. Based upon these calculations, the model then computes the total probability-of-kill. An engagement ends upon successful intercept or the missile flying past the target, impacting the ground, or self-destructing.

Model Conventions

Due to the complexity of the *ESAMS* model, this subsection provides only a subset of the assumptions and limitations associated with using this model. See the *ESAMS* Analyst Manual including the classified appendices for more details.

ESAMS reflects the assumptions upon which its input data for radar guidance and control, autopilot, thrust, aerodynamic, and warhead data are based. These assumptions are listed below under the heading Radar, Guidance and Control, Endgame, and Other:

Radar

a. Only one radar cross section (RCS) is allowed per flyout. This means that only the tracking radar or illuminator frequency can be used. The effect of bistatic RCS for the seeker is not represented.

b. The RCS is assumed to be laterally symmetric, i.e., the left half and the right half of the air vehicle are identical.

c. The RCS signature is a point source. Near-field effects are only modeled for fuzing.

d. The ‘native’ multipath and clutter model assumptions are detailed in the “GRAM Analyst Manual.”

e. The GRACE multipath model does not use triangular digitized terrain directly; instead, it uses a coarse-grained representation in which an illumination patch is made of two planar segments which have no transverse tilt. The first segment runs from the true location of re-illumination point to the true location of the midpoint of the patch; the second segment runs from the midpoint to the true location of the masking point ending the patch. At present, only the second segment contributes to multipath scattering.

f. The clutter model uses the terrain type only. It does not use the tilts associated with the digital terrain.

Guidance and Control

a. A four step Runge-Kutta integration is used (for most systems) with a fixed time step. There is no internal checking on the accuracy of the results. The time step has been chosen to generate stable results under the expected regime of input data.

b. The autopilots use an integration methodology that matches the closed-form response of the transfer functions to a step-sampled input. It is, in effect, a type of Z-transform approach, rather than a classic Euler integration.

c. Missile motion is simulated with a five degree of freedom (5-DOF) model, which computes profiles for trajectories based on positional changes in X, Y, and Z and angular changes in pitch and yaw. 5-DOF assumes a constant missile roll angle.

d. The commands uplinked and downlinked from the missile are assumed to be free of communications channel induced errors.

End Game

a. End game calculations assume equivalent dimensions of height and width of vulnerable aircraft components in determining the probability of kill due to warhead fragmentation. As a result, when building a presented area table, one should place a constant value (equal to the product of the component's length and width) in all positions. *ESAMS* only uses presented area table data to compute the amount of vulnerable area in the spray, ratioing the presented area based on the aspect angle at intercept. See the "*ESAMS* Analyst Manual-Basic Methodology" for more details.

b. Fuzing is generally dependent on the specification of glitter points and a fixed delay time. The requirements of some fuzes to accumulate energy to some level before detonation is only modeled when the detailed fuze model is selected. As a result, caution should be exercised when using Pk data. This is especially true for high speed intercepts and intercepts with small targets or targets with small signatures.

c. The closest point of approach (CPA) is calculated based on the target and missile position and velocity at fuzing or at a simulation-ending condition. Hence, caution must be exercised when using the CPA data, especially for maneuvering targets and where the miss-distance is large.

Other

a. The digital terrain facets will, in general, exhibit slope discontinuities at the edges if they are tilted. See the "*ESAMS* Analyst Manual" for more details on terrain data.

b. The maximum number of flight path points is currently limited to 1200. This is changeable at compilation time.

c. The maximum number of replications when using Monte Carlo replications is limited to 50.

d. A maximum of 10 terrain files can be input by the model. A maximum of 175 terrain squares (of 40 x 40 points) can be input.

e. The maximum number of sites is 500.

Logic Flow Through Major Components

Figure 1 shows the high-level logical flow in *ESAMS*. The major components of *ESAMS* are the RF sensor and missile flyout. The RF sensor includes acquisition and tracking modes of the target tracking radar. The flyout includes launch/no-launch decision criteria and missile flight dynamics and control. Endgame includes fuzing and damage calculations. Countermeasures may be invoked anytime during early stages of acquisition, tracking, and during the flyout stage of simulation.

RF Sensor

The *ESAMS* simulation starts with the tracking radar in acquisition mode. The sensor is "perfectly cued" to the target. In this mode the radar antenna is kept pointing directly at the flying target until it signal is detected above the noise and clutter levels. This does not correspond to a normal operational radar system mode. This mode determines the longest range at which the target can be detected, it therefore represents the maximum capability of the SAM system or "worst case" scenario for the target aircraft.

The acquisition mode has logic built in to terminate processing if the target is not acquired after a user specified time (i.e., "try again" diamond). This reduces processing time when signal is not sufficient to acquire. After successful acquisition of the target penetrator, the sensor then switches to the track mode of the tracking radar. The radar attempts to track the target in azimuth, elevation, and range. If track is established and track maintenance is lost (i.e., first "track" diamond), the track sensor goes into a "coast" mode for a user specified period of time while attempting to re-establish track (i.e., second "track" diamond). If track updates are not found within that time-frame, track is broken and the RF sensor tries to re-establish acquisition. If track is maintained until launch conditions are met, this generates a target lock-on which serves as the transition to the missile flyout portion of the simulation.

Missile Flyout

After target lock-on is established, *ESAMS* determines if the SAM system can fire at the target penetrator. Before commencing its firing sequence, the site must meet specified launch criteria, including target range, velocity, and signal-to-noise ratio (SNR). Meeting these criteria, the site begins its firing sequence and launches a single missile. For a short time the missile boosts, then guidance begins. Ground or onboard commands are generated to guide the missile toward the target.

The missile attempts to close with the target and activates the fuze. Upon fuzing, the model calculates the probabilities-of-kill from both warhead blast and fragmentation. The probabilities are based on warhead fragment characteristics, including fragment size and velocity, the aircraft vulnerability to the fragments, and the aircraft vulnerability to the warhead blast. Based upon these calculations, the model then computes the total probability-of-kill. An engagement ends upon successful intercept or the missile flying past the target, impacting the ground, or self-destructing.

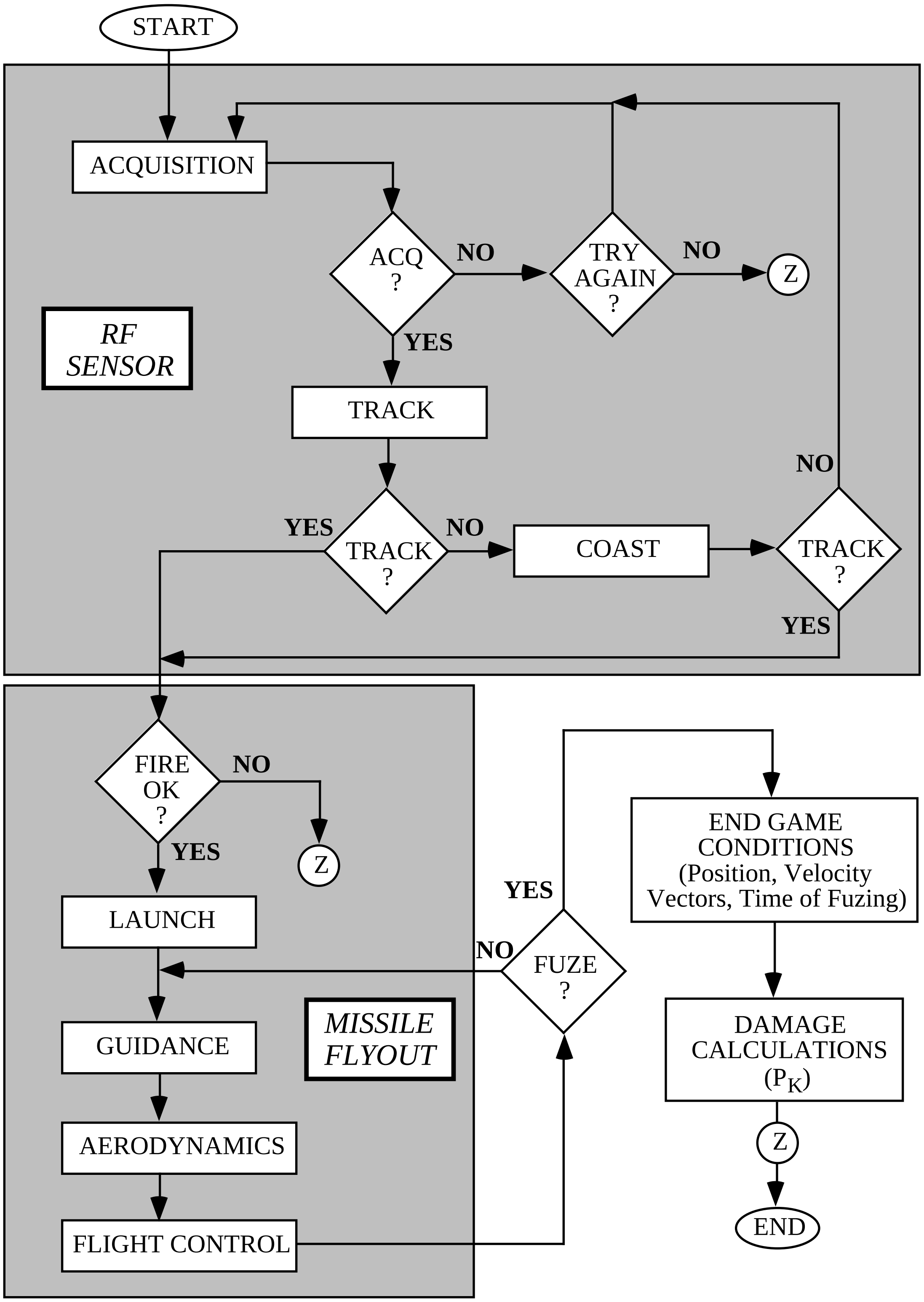

FIGURE 2.0-1. Logic Flow Diagram.

Data Flow Through Major Components

Figure 2 shows the high-level data flow for *ESAMS*. It shows the common input data files on the left flowing to the major components in the middle and then on to the output files on the right.

The Basics. The *ESAMS* model consists of software processing components and a simulation data base of pre-specified files containing missile, target, and environmental characteristics. The *ESAMS* user has access to this data base in performing the various software processing activities of scenario preparation, scenario execution, and simulation output analysis. This section presents a general overview of *ESAMS* input requirements.

ESAMS Input Data. Each of the major software components (Preprocessor, Simulator, and Postprocessor) require two kinds of input: a group of data files, and a program control (PROGC) command set. Contents of the data files depend on the software components being run. Both the Preprocessor and the Simulator process data files containing missile, target, and environmental information. The Postprocessor component requires input data files containing Simulator-generated events. The Control Command set directs the software components to perform specific operations during execution. Each of the software components has its own Control Command set.

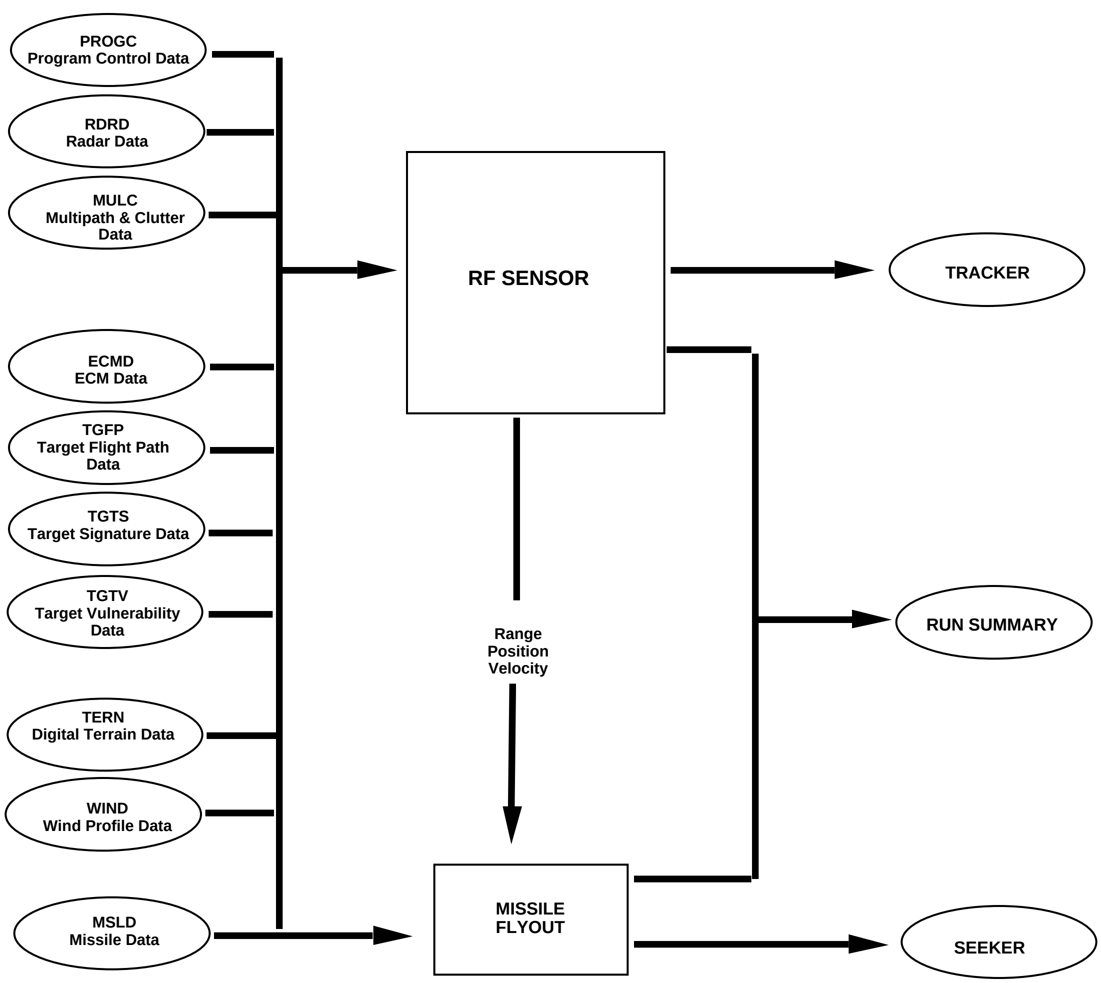

FIGURE 2.0-2. Data Flow Diagram.

Model Internal Structures. Internally the *ESAMS* Preprocessor and Simulator access information from a set of nine common blocks: RDRD, MULC, ECMD, TGFP, TGTS, TGTV, TERN, WIND, and MSLD. These storage areas comprise the simulation data base covering radar characteristics (RDRD), multipath and clutter data (MULC), electronic countermeasures (ECMD), target flight path data (TGFP), target signature parameters (TGTS), target vulnerability data (TGTV), environmental effects of terrain (TERN) and wind (WIND), and missile characteristics (MSLD). Data file inputs fill these common block areas upon execution of the particular software component. And, importantly, the structure of most data file inputs are images of these common blocks.

ESAMS Output Data. There are three output files for *ESAMS*. The primary output file is the Run Summary file which reports data from both major components of *ESAMS*. It provides the summary of acquisition, track, and target position information from the RF Sensor and the launch conditions, missile position and aerodynamic information from Missile Flyout and endgame calculations including probability of kill and summary statistics. Detailed tracker information is output from the RF sensor component to a tracker file for further analysis. Detailed seeker information is output from the missile flyout component to a seeker file for further analysis.

Source Code Hierarchy

The source code hierarchy for the major components of *ESAMS* is shown in Figure 3. The program ZINGER invokes the missile simulation executive routine called SAMS.

Subroutine Descriptions

Subroutine SAMS provides top-level control when running ESAMS for multiple engagements such as multiple sites, multiple shots per site, and/or multiple Monte Carlo replications per shot. The control for each individual engagement is managed by subroutine GORUN which calls the appropriate routines for sensor modeling (IREXE or RADAR), missile launch and flyout (AIMPT, LAUNCH, CONSYS, MISIL) or the ESAMS gun simulation (SHOOT), and endgame modeling (ENDER).

Radar modeling in ESAMS 2.7 uses either a waveform-driven methodology in subroutine WFSYNC that can model any number of different radar modes including acquisition and track modes with and without PRF staggers or a time-stepped methodology for track-while-scan (TWS) radars in subroutine TWSYNC. Radar acquisition modeling is performed in subroutine WFAPDT, track modeling in WFTCPI, and seeker modeling in SKRCPI.

Missile modeling is controlled by subroutines CONSYS and MISIL. CONSYS calls the threat-specific guidance routines, and MISIL calls AUTOP to compute threat-specific autopilot responses to the guidance commands, AERO to obtain missile airframe response (force and moments), and ACCEL to integrate the equations of motion and update the missile state.

ENDER checks for missile endgame conditions after each missile update and invokes the endgame evaluation driver called ENDGM at the appropriate time. The check for missile fuzing is performed by subroutine FUZING, and when fuzing has occurred, PKILL determines the probability of target kill based upon target vulnerability to blast and fragment flyouts.

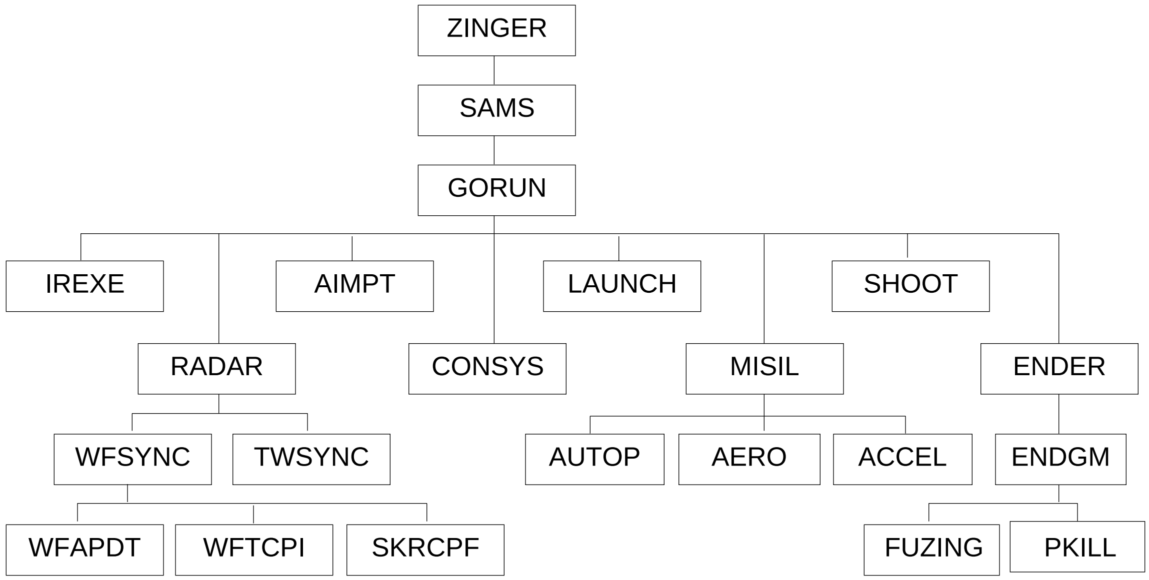

FIGURE 2.0-3. ESAMS Source Code Hierarchy.

High-Level Requirements

ESAMS shall be a digital simulation of a one-on-one encounter between an airborne vehicle and a surface-to-air (SAM) missile system.

ESAMS shall provide probability of target kill as its primary output, and shall enable the user to examine details of the engagement such as the missile flight path, guidance characteristics, and the effect of ECM and terrain on an engagement.

ESAMS shall simulate all weapon subsystems plus electronic countermeasures (ECM) for the SA-2 through the SA-16, SA-19, SA-N-1, 3, 4, 6, and 7, and the CADS-1.

ESAMS shall provide a user capability to create simple penetrator flight paths.

ESAMS shall accept flight path data created by external flight path generators, specifically Blue Max II.

RF Sensor

ESAMS shall simulate an RF sensor in acquisition, tracking, and illuminator modes.

ESAMS shall calculate the minimum signal required for acquisition, track establishment, and track maintenance measured by the SAM radar as a function of many factors (atmospheric transmittance, multipath and clutter, terrain masking, etc.), and countermeasure interference.

ESAMS shall simulate countermeasures during engagement.

Missile Flyout

ESAMS shall simulate a missile system with launch, aerodynamics, guidance and control, and fuzing characteristics modeled.

ESAMS shall account for detection range, delay time (of enabling fuzing), and cutoff range in the fuzing simulations.

ESAMS shall utilize gimbal limits and Doppler limits as well as detection range in the simulation of the missile seeker.

ESAMS shall require that the site meet specified launch criteria, including target range, velocity, and signal-to-noise ratio (SNR) before commencing its firing sequence.

ESAMS shall launch a single missile that boosts for a short time before ground or onboard commands are generated to guide the missile toward the target.

ESAMS shall provide for simulation of user-selectable instrumental and environmental noise processes, as well as the effects of various types of countermeasures during missile flyout.

ESAMS shall calculate the probabilities-of-kill from both warhead blast and fragmentation.

ESAMS shall end the engagement upon successful intercept or the missile flying past the target, impacting the ground, or self-destructing.

Implementation Requirements

ESAMS shall be written in ANSI Standard FORTRAN 77.

ESAMS shall be compatible with the following computers/operating systems:

IBM 3090 / VM/CMS
VAX / VMS
SUN / UNIX

2.0.5 DETAILED SOFTWARE DESIGN

The following sections contain detailed software design specifications for RF Sensor and Missile Flyout FEs that correspond to those of the FAT in Appendix A. The sections are numbered according to their order of appearance, but the FE designators that appear at the top of each page correspond to those in the FAT. Decomposition of the *ESAMS* model into generic, identifiable FEs that correspond to real world RF SAM systems, their targets, and the environment formed the basis for the FAT, which provides a framework for reporting results and facilitates comparisons of functionality among similar models.